National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ERB-986

Witness: A System For Object Recognition Using Range Images

C. Archibald and M. Rioux

January 1986

Canada

NRCC No. 25588

86-04295/

Copyright 1986 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1986 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

WITNESS: A SYSTEM FOR OBJECT RECOGNITION USING RANGE IMAGES

J.C. Archibald and M. Rioux

ERB-986
NRCC No. 25588

JANUARY 1986

TABLE OF CONTENTS

		Page
ABSTRACT/RÉSUMÉ		v
I.	INTRODUCTION	1
II.	RELATED RESEARCH	2
	Pixels, Voxels, and Surfels	2
	Edge and Region Detection in Range Images	3
III.	THE LASER RANGE FINDER	5
IV.	PLANAR SURFACE EXTRACTION FROM RANGE IMAGES	7
	Gradient Space	8
	Gaussian Smoothing	8
	The Gradient Operator	8
	Clustering	10
	Merging the Regions	11
	Verification of Surfaces	11
V.	MODELING THE OBJECTS	13
	Surface Properties	15
	Surface Adjacency Properties	16
	Generic View Model Properties	16
	Adequacy Issues	16
VI.	MODEL MATCHING	17
	Filter the Knowledge Base	18
	Heuristic Augmented Graph Matching	18
	Reliability Constraint	18
	Structural Constraint	19
	Relational Constraint	19
	A Scoring Mechanism for the Match	20
	Compare Nodes	20
	Compare Links	21
	The Matching Algorithm	21
	The Deduction	21
	Driver	22
	Matcher	22
VII.	RESULTS AND EVALUATION	23
	CPU Time Requirements	23
	Image Features Detected	24
	Object Recognition Statistics	24
VIII.	EXTENSIONS AND CONCLUSIONS	25
REFERENCES		25

TABLE OF CONTENTS (Cont'd)

Page

FIGURES

1 Range finding technique using synchronized geometry . . . 6
2 Two-dimensional Gaussian function 9
3 First partial derivative of the Gaussian function 10
4 (a) Clustering neighbors. (b) Merging neighbors 10
5a A range image displayed as intensity. The light areas are nearest the range finder 12
5b An oblique view of the data in Fig. 5a 12
5c The panar surfaces detected in the range image shown in Fig. 5a 13
6 (a) Sketched view of truncated pyramid. (b) Surface adjacency structure for truncated pyramid sketched in Fig. 7 14
7 Structure of the knowledge base 15

ABSTRACT

The development of new data acquisition devices demands new methods of data interpretation. This paper describes a new device capable of acquiring accurate range images in real time, and details methods which have been developed to use this range data in a robotics environment. A new approach to object recognition for a robotics environment is presented. Two-dimensional models enriched with three-dimensional information are constructed automatically from a range image. These "view models" are used to recognize objects by matching them to models subsequently constructed from similar images. A segmentation method for extraction of planar surfaces from range images has been developed. View models are constructed as augmented surface adjacency graphs. Heuristic model matching methods are presented. Results indicate that object recognition in a constrained domain and environment can be achieved at nearly a 100% success rate using single view range images.

RÉSUMÉ

La mise au point de nouveaux dispositifs d'acquisition de données nécessite l'utilisation de nouvelles méthodes d'interprétation des données. La présente communication décrit un nouveau dispositif permettant l'acquisition en temps réel d'images de distance précises et explique les méthodes qui ont été mises au point pour l'utilisation de ces données de distance par des robots. On présente une nouvelle approche pour la reconnaissance des objets par des robots. Des modèles bidimensionnels enrichis d'éléments tridimensionnels sont construits automatiquement à partir d'une image de distance. On utilise ces "modèles de vision" pour reconnaître des objets en les appareillant à des modèles construits ultérieurement à partir d'images similaires. On a mis au point une méthode de segmentation pour l'extraction de surfaces planes à partir d'images de distance. Les modèles de vision sont construits sous forme de graphiques de contiguïté de surfaces augmentés. On présente des méthodes heuristiques d'appariement de modèles. Les résultats montrent que la reconnaissance des objets dans un domaine et un environnement restreints peut être réalisée avec au taux de réussite de presque 100% lorsqu'on utilise les images de distance suivant une seule orientation.

WITNESS: A SYSTEM FOR OBJECT RECOGNITION USING RANGE IMAGES

Colin Archibald and Marc Rioux

I. INTRODUCTION

The availability of range data simplifies the problems associated with object recognition. Faster and more accurate devices are being developed rapidly. The camera used in this research is capable of capturing depth information in real time. This new device demands new data interpretation techniques. (For an informative survey of range finding techniques, see ref. 1.)

In our simulated robotics environment, the field of view for the range finder is a portion of the work envelope of the robot arm. The range finder is directed straight down. The objects to be recognized are all polyhedra. The problem is easily defined: How can the single view range image be used to recognize the objects in the robot's work envelope?

In this paper we describe a method for collecting features from a particular view of an object. These features are placed in a data structure called a view model.

A view model consists of topological and geometric properties of the surfaces that are visible. An amalgamation of these properties is also a part of the view model and is referred to as generic knowledge. There can be several view models for each object, one for every stable resting position. View models which are orientation independent with respect to that particular view are taught to the system automatically. These view models form the knowledge base for the domain and can be maintained interactively. The same methods that are used for building the knowledge base are used subsequently to build view models for recognition. Because of this we have chosen to call the system "Witness," a system that recognizes what has already been seen. In the recognition stage, view models are created for the objects to be recognized in the scene. These models are compared to the models in the knowledge base, first generically to obtain a list of possible matches, and then a more elaborate matching algorithm is used to determine which of the possible models best matches the object to be recognized.

In much computer vision research, a two-dimensional image has been used with the intention of extracting a three-dimensional model (2). In these systems, depth cues, such as shadows and perspective, are interpreted. In our research the modeling method used is basically a two-dimensional description of what is in the three-dimensional scene. We believe that, to

achieve a success rate of 100% for object recognition, no guessing of hidden surfaces or other invisible properties of the scene should be allowed. Using range images, extraction of information which is explicitly available in the image is sufficient for object recognition in a constrained domain.

In the following section recent research in range data interpretation is discussed. Section III is a description of how the range image is acquired, and Sections IV, V, and VI describe the implementation of Witness. Experimental results are evaluated in Section VII and some possible extensions are presented along with conclusions in Section VIII.

II. RELATED RESEARCH

Pixels, Voxels, and Surfels

Most computational vision systems use images represented as a two-dimensional array of values called pixels (picture elements). These pixels may consist of hues, grey levels, or binary values. With the advent of computer aided tomography, used mainly for medical imagery, a new type of image processing was introduced (3). This type of image is a three-dimensional array of intensities, representing density. The elements of this array are called voxels (volume elements) and must be treated differently during analysis. The camera described briefly in Section III is one of the recently developed sensors referred to as "range finders." The images produced by these devices are two-dimensional arrays of depths, i.e., the distance from the camera to the surface being scanned. These elements will be referred to as "surfels" (surface elements), and must also be processed using methods developed specifically for this type of image.

The benefits of using range images are obvious to those who have attempted to extract depth cues from two-dimensional images to obtain feasible three-dimensional interpretations. Range images provide the three-dimensional information explicitly, which eliminates the costly and unsure task of implying depth in a scene. A range image is easier to analyze than an intensity image because the information in this type of image is a function of only one scene property, namely depth. When processing an intensity image, factors such as lighting, surface reflectivity, and surface orientation must be considered, complicating what is already an inadequate source of three-dimensional information. However, a range image can only be acquired for surfaces within a limited range of distances from the range finder. This restriction is not a serious problem in the indoor factory environment where the objects occur within the constant and short range of the range finder.

This paper is concerned with both low level and high level processing of a range image. The purpose of low level image processing, or image analysis, is to compute a description of the image to be used for subsequent processing. In high level processing, or scene analysis, the image description is used to determine what is in the scene.

With the potential benefits of successfully exploiting range data for robotics, a number of interesting papers have appeared. The methods used to interpret range data, and indeed the devices themselves, vary greatly with the task.

Edge and Region Detection in Range Images

Generally, object recognition from images is accomplished by extracting features of the object from the image and matching these to previously defined models, or collections of features pertaining to an object. An image is represented as a very large array of elements from which image features must be identified. The first stage in image processing is to manipulate the image into a manageable form which will make these features more apparent. As in intensity image processing, low level processing of range images is either "edge based," "region based," or some combination of the two.

Edge Detection

A different approach is used for edge detection in range images from that used for intensity images. Different information is available from surfels than from pixels. Edge information, derived from range images, often consists of both location and type of the edge. There are three types of edges commonly identified. If the two surfaces form a convex intersection, the edge where the surfaces meet is referred to as a convex edge. Concave edges are defined similarly. The third type occurs when a surface is occluding some part of the scene. The edge formed here is characterized by an abrupt change in range values, and is referred to as a jump edge, or jump boundary.

Nitzan et al. (4) extract jump boundaries from range images to find outlines of objects in the scene. This information is used to compute the Cartesian coordinates of the outline of the object, and obtain normalized views of the interesting surfaces. Mitiche and Aggarwal (5) extract edges from a noisy image using a model driven algorithm. The probability that an edge exists at each surfel is used to help derive an edge map for the image. Inokuchi et al. (6) use a combination of edge and region detection. They have developed an operator which is used to label every surfel as being on a jump edge, convex edge, concave edge, or planar region. Gil et al. (7) detect edges in registered intensity and range images, and combine these edge maps.

Region Detection

Extraction of planar and quadratic surfaces is often done by histograming techniques, as described by Duda et al. (8). Different histograms are used to identify horizontal, vertical, and slanted surfaces. Planar surfaces are approximated from the histogram and refined by fitting a plane to this section of the image. Reliable surfaces are identified first, and then questionable ones are attempted.

In ref. 9, Hebert and Ponce mapped the image into a two-dimensional space, namely surface normals represented on the Gauss sphere. In their method, the planar, cylindrical, and conic surfaces are detected globally using a Hough transform, and distinguished by location in the image plane. The "Three-Point Seed" method of planar surface extraction is described by Henderson and Bhanu (10). Three neighboring surfels are selected, and if they approximate a plane, the algorithm will grow this surface to its edges. The surfaces are approximated as polygons and used to create a model of the object.

In this research we favor region based segmentation methods. When extracting information from a range image or any other digital signal, it is best to use as much of the signal available as possible. Algorithms which extract regions from an image are using more of the surfels to build a region map than would be used in building an edge map. The region map is very likely to be more reliable because more correct information is used.

Of course, region detection can only be used if there is a constant measure over the region to be found. In range images of a polyhedral world, surface slope can be extracted and used as such a measure.

There is another significant reason to favor region detection over edge detection in this domain. A slightly incomplete region will be less harmful to subsequent processing than an incomplete edge map. Using incomplete edges to build an object model is a very difficult task and will inevitably result in models which are not as reliable as the images are capable of providing using a region based method.

Modeling and Object Recognition

Model based vision systems have been using intensity images since 1965. (For a complete summary of this work see ref. 11, Chapt. 3.) Several methods of object modeling have been proposed. One way is to create a dictionary of primitives and represent the objects in terms of these simple objects. This approach was used by Roberts (12), who used simple polyhedra as primitives. An assembly of generalized cylinders and ways in which they

can be connected are used in Nevatia and Binford (13). Generalized cylinders have also been used in Popplestone et al. (14). This representation is ambiguous in that an infinite number of cylinders can represent a single object.

Modeling of polyhedra has been a popular place to begin research in object recognition. Falk (15) uses topological and geometric properties of objects in a representation called face adjacency graphs. The nodes in these graphs represent planar faces and the links represent connecting edges. The graphs are used to recognize objects which are found in incomplete and imperfect line diagrams of the polyhedra. Falk's objective is quite ambitious. From a single view line drawing, he attempts to identify objects which may be in any position and orientation and also may be partially occluded by other objects in the scene. Although Falk's methods are sound, a limited amount of information can be extracted accurately and consistently from a single two-dimensional image.

Oshima and Shirai (16) have used the concept of face adjacency graphs object recognition using range images. Their goals are also ambitious, as they attempt to recognize polyhedra that may occlude one another in varying degrees from a single view range image.

Much research in range data interpretation is based on the requirement that a robot vision system must recognize an object in any position or orientation. This has resulted in attempts to match a partial model created from a single view to a complete three-dimensional model. Henderson and Bhanu (10) attempt to build a complete three-dimensional model by rotating the object on a platform, and taking several range images. Others have indicated that a CAD/CAM type model will eventually be used in their systems. (Shirai (17) suggests this as a solution.)

It is a contentious issue as to what restrictions need to be placed on the object domain, and the working environment, to achieve a performance rate for object recognition that is suitable for robotics applications.

III. THE LASER RANGE FINDER

A new tool for 3-D acquisition has been designed and implemented during the past two years. It is a laser range finder capable of real-time reading of surface topography of a scene. The device is based on triangulation, but we modify the usual geometry in order to increase the speed of acquisition and decrease the volume of the optical head, as well as the importance of shadow effects.

The basic idea is to synchronize the projection of a light spot with its detection. This is realized using two scanners instead of one. The first scanner projects a beam on the surface to be measured and the second one is coupled to the receiving sensor in order to "follow" the spot. Two distinct scanners can to used to provide projection and detection but it requires critical adjustments in order to get stable synchronization. On the other hand, perfect synchronization can be obtained using polygonal or pyramidal mirrors. The idea is to use one facet of the scanner to project the light beam and one other facet of the same scanner for detection. Among the numerous geometrical arrangements, the one depicted in Fig. 1 has been implemented in our laboratory. It uses a pyramidal mirror consisting of six facets. A laser beam from source S is deflected by a facet of the scanner to a fixed mirror M1. A second deflection occurs on the deflecting mirror M3. After scattering of the light onto the surface of the object, the beam is deflected again on M3 to another fixed mirror M2. The opposite

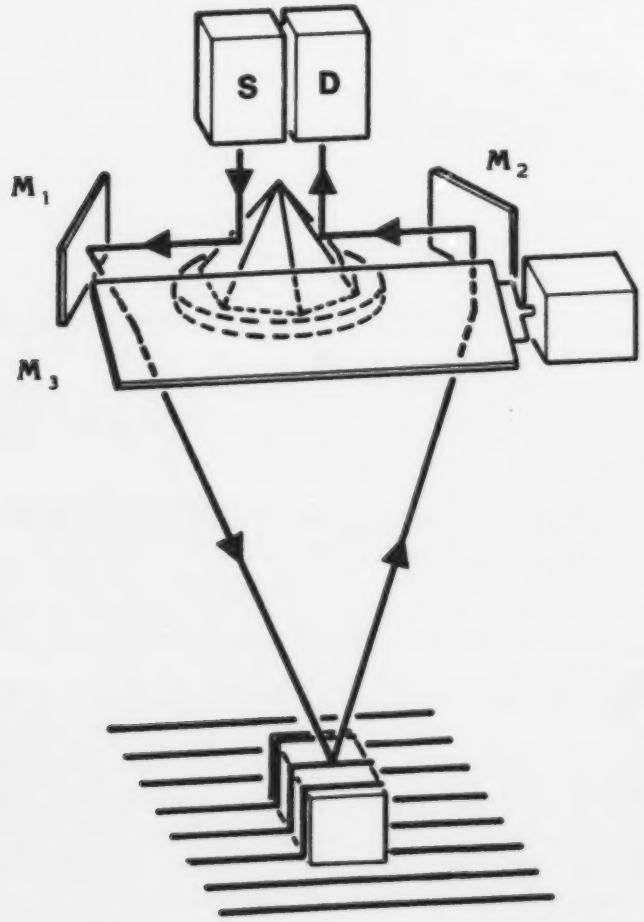

Figure 1. Range finding technique using synchronized geometry.

facet of the pyramidal mirror is then used to synchronize the detection beam. A position sensor D provides the profile measurement.

One interesting property of a synchronized approach is that it requires a linear position sensor only (in contrast to usual geometry), which increases the bandwidth of the profile signal from ~10 kHz to 1 MHz. The position sensor used here is a lateral effect photodiode. It consists of a photodiode deposited on top of a resistive layer. When a light spot is imaged on the surface of the photodiode, the current produced is split by the resistive layer so that the position information can be simply obtained by dividing the difference of the currents by their sum. The response time of the position sensor is 500 ms. Sampling of the position signal is made at a rate of 600 kHz. Notice that the sum of the current is proportional to the amount of light diffused by the surface in the direction of the camera. We sample synchronously that signal in order to get an intensity image (similar to what a TV camera would provide), that is in perfect registration with the three-dimensional image.

Our experimental prototype uses a diode laser as a source. The pyramidal mirror scanner consists of six facets of ~5 cm^2 and is rotating at a rate of 166 revolutions per second which provides 1000 profiles per second. In order to increase the signal-to-noise ratio, we average six profiles which allows us to capture a scene of 256 × 128 pixels within a second. Resolution along range (z) is smaller than a millimetre for a volume field of view of 250 mm (x) × 250 mm (y) × 100 mm (z).

A more detailed description of this device can be found in refs. 18 and 19.

IV. PLANAR SURFACE EXTRACTION FROM RANGE IMAGES

All planar surfaces have constant x and y slopes, but only surfaces parallel to the image plane have constant range values. To identify all planar surfaces in the image using the same algorithm we have chosen to compute the gradient space coordinates for each surfel from the Cartesian coordinates which are explicit in the range images. In this representation all planar surfaces can be detected by local consistency of both x and y slopes.

The range images contain a small amount of noise which must be removed, or at least be made insignificant, before surfels can be considered individually. An operator is introduced which when convolved with the image results in a directional gradient value while simultaneously smoothing out high frequency noise.

Gradient Space

The gradient space (G), introduced by Mackworth (2, 20), is a coordinate system which facilitates the mapping of a three-dimensional scene into two dimensions. The axes represent x slope (usually labeled p), and the y slope (usually labeled q). When a planar surface is mapped into G, it is represented as a point. More concisely stated: If a plane is defined by $Ax + By + Cz + D = 0$, it is mapped into G as $(p, q) = (-A/C, -B/C)$.

Gradient space has been used extensively in image analysis and has many interesting properties. A detailed description of gradient space and its applications can be found in ref. 21.

Gaussian Smoothing

The range images used in this research were found to have very little noise. However, experimentation has shown that some smoothing would improve results. Marr and Hildreth (22) have shown that local averaging based on the normal distribution can be used to create a low pass filter which is optimal in the tradeoff between spatial localization and maintaining a small variance in the frequency domain. In our case this low pass filter can be used to smooth high frequency noise from the range images. The one dimensional normal distribution function, often called the Gaussian function is:

$$G(x) = (2\pi\sigma^2)^{-1/2} \, e^{-x^2/2\sigma^2}$$

Hereinafter, the scale factor will be ignored. In two dimensions, without the scale factor, the function is represented as:

$$G(x, y) = e^{-(x^2+y^2)/2\sigma^2}$$

This function is represented graphically in Fig. 2. Our objective is to smooth the image and approximate the first partial derivative at each surfel in the image. An operator designed to perform these two tasks simultaneously is described in the next section.

The Gradient Operator

Gaussian smoothing in two dimensions is implemented by creating a square matrix called a mask, which is a discrite representation of the normal distribution. The mask is used to convolve the image with the Gaussian function.

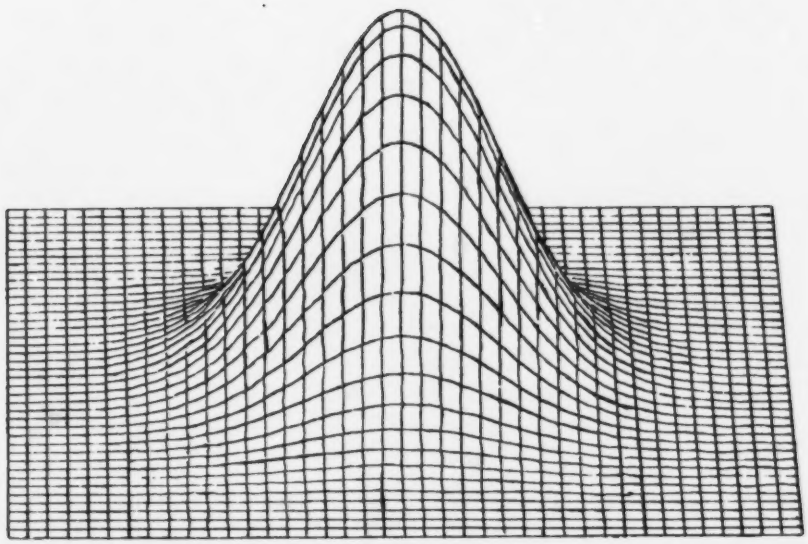

Figure 2. Two-dimensional Gaussian function.

Notice that this function is rotationally symmetric, and therefore non-directional. Taking the first partial derivative of the Gaussian function with respect to x results in:

$$\frac{\delta G(x, y)}{\delta x} = \frac{-x}{\sigma^2} e^{-(x^2+y^2)/2\sigma} = \frac{-x^2}{\sigma^2} G(x, y)$$

This function is represented graphically in Fig. 3.

There is one parameter, σ, which should be determined carefully when using real world images. In our controlled environment, approximately the same amount of high frequency noise is apparent in every image, thus σ was selected simply by visual examination.

When applied to a range image at a point P, the resulting value is an approximation of the slope in the x direction at P. A value to approximate the y slope at P can be determined by rotating the mask 90°.

This operator is similar to difference operators applied to intensity images by Sobel, Kirsch, and others (23) with the added advantage of noise removal. It also returns values of large magnitude at points in the image where a jump edge is present. This is not surprising, since the operator is a discrete approximation to first-order partial differentiation.

A planar surface extraction algorithm based on local consistency will not ignore these jump edges.

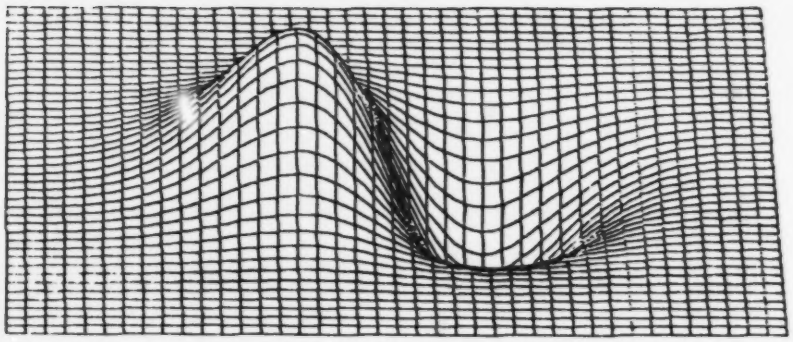

Figure 3. First partial derivative of the Gaussian function.

Clustering

The first stage of surface extraction is a single top-to-bottom scan of the x and y slope images simultaneously. During this scan neighboring surfels which are similar in both x and y slope are clustered into a region and given a label. "Similar" means to be within a threshold. (This threshold is currently set at ~15°.) For the clustering algorithm the first row and column of the region map are defined to be region label 1. The x and y slope of each surfel S (excluding the first row and column) are compared to the x and y slope of its four preceding neighbors in the scan. These neighbors are denoted N1 - N4 in Fig. 4a. If S is not similar to any of these neighbors, it is assigned a new region label. If it is similar to one or more neighbor, it is assigned the label of the neighbor with the minimum total magnitude difference in slopes.

In an 8-connected manner, but considering only four neighbors, the scan proceeds to create a region map, and a data structure containing the number of surfels in each region, and the average x and y slopes for that region.

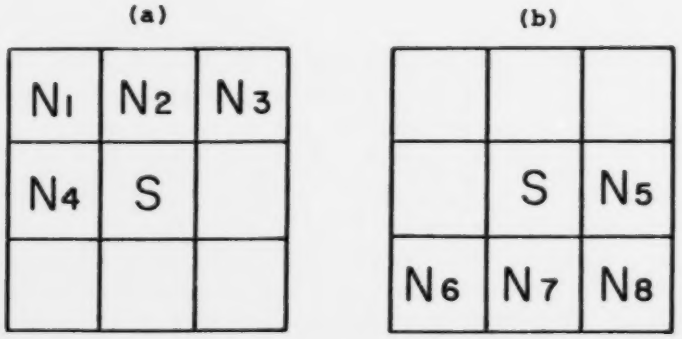

Figure 4. (a) Clustering neighbors. (b) Merging neighbors.

This scan cannot result in an adequate representation of surfaces in an image. Consider a U-shaped surface. This top down scan would at best result in two adjacent regions representing that surface.

The purpose of this scan is to reduce the image to a region map and a data structure. Although there may be 500 or more of these regions, varying with image content, merging the regions will certainly be less computationally costly than merging at the surfel level.

Merging the Regions

In this stage the region map derived above is scanned top to bottom repeatedly until all adjacent, similar regions are merged. At each point in the region map, the region label of the succeeding four neighboring surfels is checked to see if there is an adjacent region present (N4 - N8 in Fig. 4b). The slopes of the adjacent region are checked for similarity, and similar regions are merged.

In the merging stage the scan is looking forward and down for adjacent regions, resulting in a complete 8-connected detection method when considered in combination with the clustering stage.

Verification of Surfaces

We have referred to clusters and regions, as opposed to surfaces, because some of the regions detected are not valid surfaces. There are two types of regions detected that do not correspond to surfaces in the scene. The first type are small regions caused by noise or the shadow effect, and these are eliminated by setting a threshold for the smallest region that will be accepted as a surface. The threshold used is approximately 160 mm. (It was found that this threshold performs well at removing unwanted regions. This is a domain and device dependent threshold that can only be determined by experimentation.) The second type of erroneous region is caused by the operator used to compute the gradients. These regions are characterized by having very large slopes. As the gradient operator passes over an edge or peak in the image, the resulting approximation to the first partial derivative is meaningless. The solution is to remove all regions with a very high slope. This was found to eliminate the erroneous regions without affecting valid long and narrow surfaces. A range image displayed as intensity is shown in Fig. 5a. Figure 5b is an oblique view of the image which gives a better appreciation for the three-dimensional nature of the data. The verified surfaces detected are indicated in Fig. 5c.

Figure 5a. A range image displayed as intensity. The light areas are nearest the range finder.

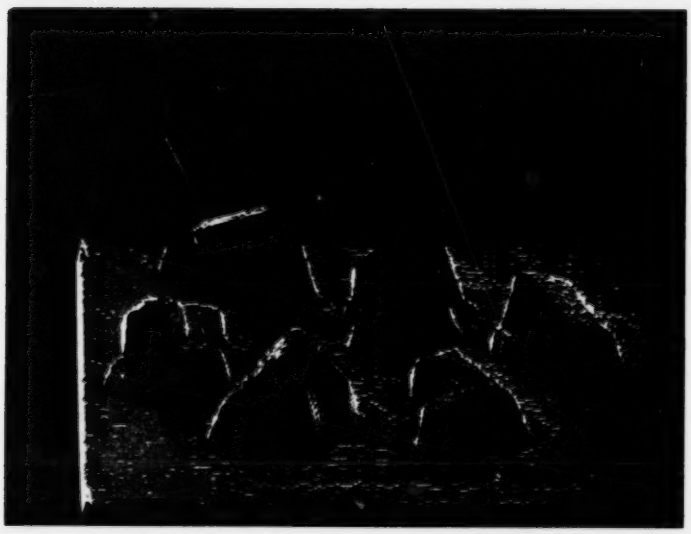

Figure 5b. An oblique view of the data in Fig. 5a.

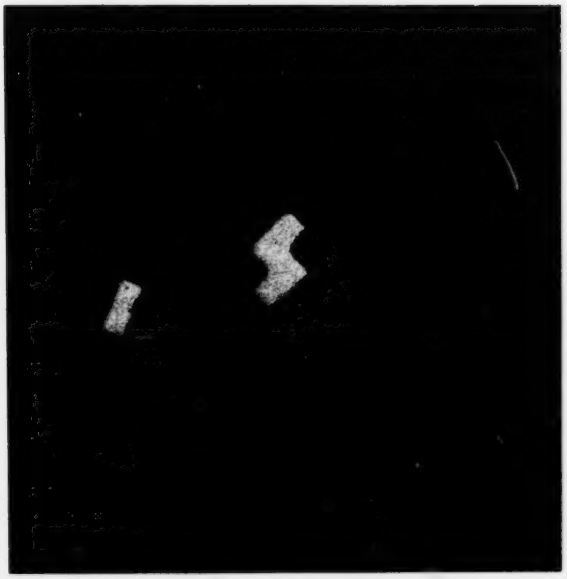

Figure 5c. The planar surfaces detected in the range image shown in Fig. 5a.

V. MODELING THE OBJECTS

A modeling scheme for rigid solids that is both descriptively and procedurally adequate (11) has been sought since the early 1970's for use in computer graphics, CAD/CAM, and scene analysis. (For a survey of these methods see ref. 24.) The proper technique to use is dependent on application and the information available for building the model. Since we are working in a robotics environment with the camera directly overhead, there is a very restricted number of possible views for each object. Each view is referred to as a resting position for that particular object. Thus, there may be several view models constructed for each object and placed in the knowledge base. The number of models required is dependent on the shape of the object. A cube, for example, has six resting positions, but only one view model is required because the object will appear the same in each position. In general, more models will be required for objects that are complex in shape. Each of these view models points to a "real world" model which contains information pertaining to the object which is true regardless of the resting position. These properties may include the object's name, weight, purpose in an assembly task, etc.

A knowledge base containing the view models is created interactively by the user in "Teach Mode." Automatic teaching of view models is an ideal

method for object recognition. The features detected by the range finder and low level processing for a particular object view are likely to be very similar to subsequent "sightings" of that object. This makes the Witness model matching process very reliable. The actual topological and geometric properties are collected automatically from the image, and the user is asked to interactively enter the non-visual object properties, if there is not already a "real world" model for that object in the knowledge base.

The framework for view modeling is surface adjacency topology. A model of a view is represented by a surface adjacency graph. Falk (15) uses a similar representation which he refers to as face adjacency graphs. To demonstrate this framework, Fig. 6a is a sketch of an overhead view of a truncated pyramid. Figure 6b is the surface adjacency graph for that object which will serve as the skeleton for the model of this view. The nodes in the graph represent the surfaces, and the links represent surface adjacencies. Putting the meat on this skeleton involves extracting the geometric properties of the surfaces and adjacencies, which are augmented to the nodes and links, respectively. All properties to be used for model matching must be independent of the object's location and orientation with respect to its particular resting position in the image. The view model for an object is used to represent only one resting position for that particular object. The model will vary slightly as the object is placed in different locations in the image due to panoramic distortion. This variation is insignificant now that a calibration procedure has been implemented to accurately convert the image to an orthographic projection.

The entire adjacency network is summarized in another structure. This object knowledge is referred to as generic knowledge as it contains information about the entire adjacency structure. The generic knowledge of the

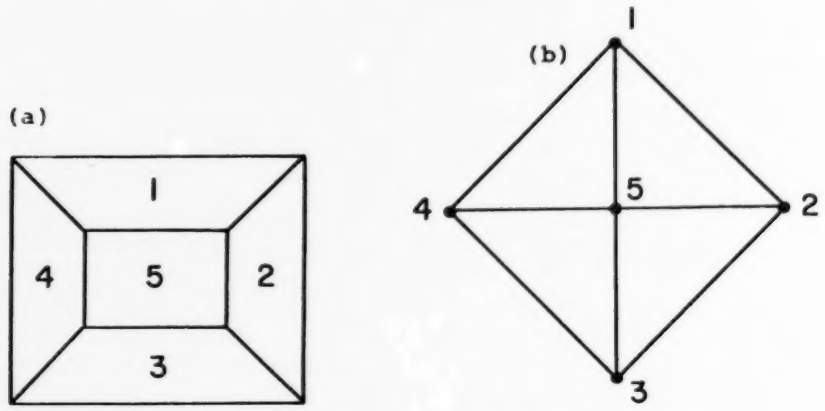

Figure 6. (a) Sketched view of truncated pyramid. (b) Surface adjacency structure for truncated pyramid sketched in Fig. 7.

model is used for a preliminary scan of the knowledge base determining which models should be considered as candidates for model matching algorithms. Figure 7 illustrates the organization of the entire knowledge base for view models.

Surface Properties

The properties collected for each surface are the following:

1. number of adjacent surfaces - determined during the trace of the object
2. projected area - measured in surfels
3. projected perimeter - measured in surfels
4. compactness = (4π projected area)/(projected perimeter)^2
5. maximum height - measured in range image levels
6. x slope = p as determined by the gradient operator
7. y slope = q as determined by the gradient operator
8. magnitude of gradient = $(p^2 + g^2)^{1/2}$
9. direction of gradient = atan (q/p)

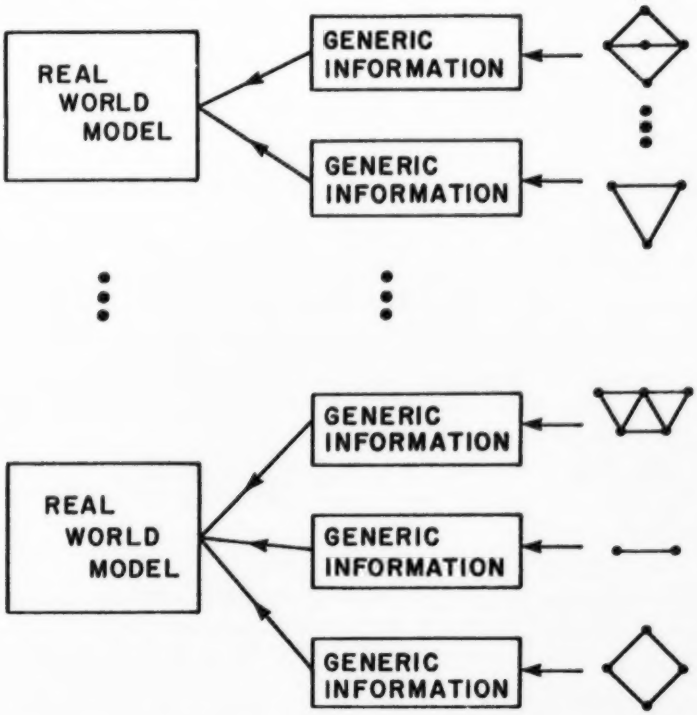

Figure 7. Structure of the knowledge base. Each view model, represented as an augmented surface adjacency graph, has generic knowledge associated with it, and points to a real world model containing non-visual information.

If an intensity image were used rather than a range image, several of these properties, such as maximum height and slope values, would not be readily available. The model is very rich, even when considering that it is only for a single view. Note that properties such as area do not represent the true shape of that surface, but the projected area seen in the image of this view as extracted by the surface detection methods.

Surface Adjacency Properties

For adjacent surfaces, two more useful properties can be extracted:

1. difference in magnitude of slope between the two surfaces
2. convexity of the surface adjacency, i.e., do the two surfaces form a convex edge, a concave edge, or does one surface partially occlude the other, resulting in a jump edge

Generic View Model Properties

Properties which typify the entire view model are collected as the adjacency structure is being constructed. They are

1. number of surfaces visible in this view
2. total projected surface area for this view
3. the maximum height of any surfel in the object

Adequacy Issues

Descriptive adequacy for view modeling in a restricted domain is a different issue from typical solid modeling adequacy criteria. The objective of this modeling scheme is not to represent an object, but to represent a particular view of an object. The difference is significant. No hidden surfaces in a particular view are included in the view model, making it a two-dimensional description of what is in the scene. The representation is made more adequate by enriching this two-dimensional description with accurate three-dimensional geometric and topologic information. Since there is no attempt to actually create an object model, the question: "What objects can be accurately represented?", becomes: "What views, and of what objects can be distinguished from each other without ambiguity?". Ambiguity in view models can occur in two ways. The first case is when two objects that are slightly different are not distinguished from one another because the range finder and low level processing have failed to detect this small difference. The degree of this ambiguity is dependent on the capabilities of the range finder and low level processing. The second case of ambiguity is when a particular view of an object is the same as a view of a different object. Although this very problem would make view modeling

impractical in most vision systems, the fact that this scheme is designed for a robotics domain makes the deficiency less serious. When this ambiguity occurs the robot can easily remove the object from the work envelope, or alter its position, so that it can be recognized uniquely.

Since there is no attempt to distinguish all solid objects, the main criteria for descriptive adequacy is application dependent. Is the representation descriptive enough to distinguish among all different objects in a domain? It is only useful, and necessary, to recognize objects which may be encountered.

The criteria for procedural adequacy of a modeling technique are more easily stated. First, the model must be designed so that all of its information is easily accessible. Second, it must lend itself to being processed efficiently. Even a model which contains all the necessary information is useless, until it can be used in the application for which it was designed.

The view model, due mostly to its simplicity, meets the first criterion very well. All of the information on all surfaces, and the object, is readily available. Pointers are used to represent the undirected adjacency graph, and each adjacency is stored twice - once with each node, for immediate graph traversal.

There is one potentially serious procedural problem with the representation in satisfying the second criteria. Since the models are represented as graphs, model matching must be some form of graph matching. Determination of graph isomorphism is easy to implement, but theoretically it provides some serious difficulties. It is still an open issue as to whether the graph isomorphism problem is NP-complete (25), so the algorithm which does this matching must be carefully guided heuristically, and should check for situations that are computationally explosive. For example, when two surface adjacency graphs have a large number of similar nodes, the matching algorithm described in the next section will be very inefficient. Our problem is complicated even more by the fact that we are not searching for an exact graph isomorphism, since the data may be incomplete.

VI. MODEL MATCHING

The model matching procedure described in this section is performed in two stages. In the first stage, the generic knowledge is used to eliminate models in the knowledge base that are not close enough to the model which is to be recognized. This stage will be referred to as "filter." The second stage is to match the surface adjacency graph of the model to be

recognized (Rmodel) to each of the models in the knowledge base that was not eliminated in the first stage. A scoring mechanism associated with this match is used to determine which of the models in the knowledge base best matches the Rmodel. This graph matching algorithm is computationally complex, and several constraints are described to guide the graph matching more efficiently.

All of the thresholds used for matching are very loose, so as not to reject any model which may have been in error in only one aspect. The idea here is to apply many loose comparisons to the various properties in the models, in order to achieve a robust overall method for matching.

Filter the Knowledge Base

The knowledge base is scanned to eliminate models which are not to be considered in the second stage of the matching procedure. A significant difference in either the maximum height of the object or total surface area is cause to reject the possibility of a match. A 100% difference in area, or 5 mm in height are the maximum differences allowed for a model to be considered in the next stage of matching.

Heuristic Augmented Graph Matching

Matching two augmented graphs is an interesting search problem. The goal is to determine if the two graphs are isomorphic and the optimal isomorphism, if such a relationship exists. The problem is further complicated by the possibility of inaccurate or incomplete data, making the concept of "almost isomorphic" meaningful. It is not practical to test all of the mappings from one graph into another. For graphs with n nodes there are n! such mappings. The search must be constrained in some manner. It is suggested in ref. 26 that invariants in the graphs, such as number of edges and vertices, be used as initial constraints. This is not a practical solution to guide a search where the data may be incomplete.

The search method used is a constrained, simultaneous depth-first search of the two graphs being compared. It determines an optimal graph match and is guided by three independent constraints: reliability, structural compatibility, and relational compatibility.

Reliability Constraint

At any stage during the depth-first search there may be several adjacent nodes which could be selected as the next node to be considered in the search. The reliability constraint guides the search to "prefer" the node which represents the surface that is most likely to be recognized in a

scene. This surface is characterized by a large surface area, low magnitude of slope, and high compactness.

The purpose of this "preference" for reliable nodes is to guide an imperfect match. All nodes will eventually be considered in a graph matching algorithm where the data used are complete, and in that case the order of the match is inconsequential. In our images a small or highly oblique surface may occasionally be missed by the low level processing. This node evaluation is used to minimize the effect of such an error on the graph matching algorithm and improve the search performance by avoiding mistakes in the models due to missing or erroneous information.

A node can receive a maximum of 100 "reliability" points. If the surface area is greater than 1000 surfels, then 50 points are awarded. If there are less than 1000 surfels, then the number of surfels divided by 20 is awarded. Twenty-five points are added in a similar manner based on the magnitude of the gradient for that surface. The compactness (which ranges from 0 to 1) times 15 is added giving a maximum total, so far, of 90 points. The remaining 10 points are determined by the number of adjacent nodes. If a node has 10 or more adjacent nodes, 10 points are added. If there are less than 10, the number of adjacent nodes is added. Although the reliability of the surface does not depend on this property, it is included to guide the search toward nodes which have many adjacencies. These nodes are preferable because the algorithm will have more adjacencies to choose from in the subsequent step.

The reliability of a node is also used to calculate the "goodness" of a match between two nodes. Matching reliable nodes adds more points to the score for a match between models than matching less reliable nodes. The mechanics of the scoring method are described later.

Structural Constraint

During the simultaneous depth first search, nodes are matched one at a time. After matching the initial pair of nodes, only nodes which are structurally compatible in the graphs will be considered for matching. For example, the second pair of nodes to be matched must both be adjacent to their respective member of the initial pair. This structural constraint is inherent in the depth first search algorithm, making it a convenient method for reducing the number of nodes which must be compared.

Relational Constraint

Once a pair of nodes has been deemed structurally compatible, they are compared to determine relational compatibility. This comparison is

computed by examining the differences in the geometric properties of the surfaces which the nodes represent. If the comparison results in a score which is above a threshold, the nodes are relationally compatible. The threshold is set at 50 points out of a maximum 100, to ensure that no possible match is missed.

A Scoring Mechanism for the Match

At each stage in the match it is necessary to know a score for the goodness of the match so far. The score is determined by evaluating the difference between the nodes and links as they are traversed. A point system where a high score represents a good match has been devised for this purpose. Properties which the range finder and low level processing found accurately are given higher weights in this point system than properties detected inconsistently. The score used for comparison of nodes is the same formula used to determine relational compatibility of the two nodes. The score for two matching nodes is multiplied by the reliability of one of the nodes divided by 10. (Since the nodes have been deemed similar, it does not matter which one is used.) This means that the matching of two reliable nodes can have up to 10 times the significance than matching two unreliable nodes in the most extreme case. This factor ensures that the matching of several small erroneous nodes will not overwhelm the score resulting from matching the reliable parts of the adjacency graph.

Compare Nodes

The comparison of nodes in the model scores the match between two surfaces. The more reliable properties are given more weight. The score is based on percent difference, with a maximum score of 100 points. The weights assigned to each property used in the comparison are as given in Table 1.

Table 1

Surface Property	Weight
Area	30
Number of adjacent surfaces	5
Perimeter	15
Compactness	15
Maximum height	15
Magnitude of gradient	20
Total	100

Compare Links

The adjacencies between the surfaces, represented in the graph by links, are already considered in the matching algorithm in a structural sense during the graph traversal. The relational properties augmented to these links should also be considered in the final score relating the two view models being matched. Fifty points are awarded here for an exact match between adjacency properties. The total score is equally divided between the convexity factor and the difference in magnitude of slope.

The Matching Algorithm

In this section, the algorithm for graph matching is summarized in pseudo-English. "Best" refers to the node with the highest score using the node evaluation function. "comp_nodes" refers to the score of the comparison of nodes, "comp_links" refers to the score of the comparison of two links, and "eval node" refers to the reliability of the surface. Recall that a "possible match" between two nodes indicates that they are in structurally compatible positions in the graphs, and relationally they have scored above a threshold using the "comp_nodes" function.

The Deduction

The reader may notice that the matching algorithm has the possible disadvantage of matching simple objects to a portion of a complex object, with a high score. For example, a view model consisting of a single surface may be matched to a complex model which has a surface very similar to the single surface model. This is corrected after the score for matching of two models is determined. The surfaces which are not used in the match are evaluated using the "eval_node" function and the total of the reliability scores of the unused nodes is deducted from the score accumulated by matching the two models.

In summary, consider an Rmodel with m nodes and a KBmodel with n nodes, where $m \neq n$. The scoring of the match between these models is as follows.

$$\text{score} = \sum_{i=1}^{\min} (\text{comp_nodes}(\text{Rmodel node}_i, \text{KBmodel node}_i) * \text{eval_node}(\text{Rmodel node}_i)/10) + \sum (\text{comp_links}(\text{all links matched}) - \text{deduction}(\sum \text{eval_node}(\text{all unused nodes})))$$

where min is the minimum of m and n.

Note that the deduction may be calculated from nodes belonging to either the Rmodel or the KBmodel depending on whether m > n, or vice versa.

Driver

Determine the "best" node in Rmodel, call it BEST.
Determine "possible" matching nodes in KBmodel for BEST.

For each "possible" matching node:

Call Depth First Search(BEST, possible KBnode, Rvisit, KBvisit, score)

Matcher

[1] Depth First Search(Rnode, KBnode, Rvisit, KBvisit, score) (Rvisit and KBvisit are arrays of nodes that have already been considered.)

[2] Visited(Rnode, Rvisit)

Visited(KBnode, KBvisit) (Record that these nodes have been considered.)

[3] score = score + comp_nodes(Rnode, KBnode) * eval_node(Rnode)/10

[4] For all of Rnode's unvisited adjacent nodes:

[5] TEMP = "best" of Rnode's unvisited adjacent nodes

[6] For all 'possible' matches for TEMP from among KBnode's unvisited adjacent nodes:

[7] Store a copy of this environment.

[8] Depth First Search(TEMP, A KBnode "possible" match, Rvisit, KBvisit, score)

(The KBnode which returns the largest score in this loop is the one to continue with - call it MATCH.)

[9] Replace the old environment stored above.

[10] score = score + comp_nodes(TEMP, MATCH) * eval_node(TEMP)/10

[11] score = score + comp_links(Rnode,TEMP,KBnode,MATCH)

[12] Visited(TEMP,Rvisit)
Visited(MATCH, KBvisit)

[13] The maximum score that appears here at the end of the recursive routine is the value of the match between these two models. This score must be saved globally.

VII. RESULTS AND EVALUATION

Performance of Witness will be measured in terms of three criteria. CPU time required for the algorithms; image features detected; and success rate of object recognition will be reported.

The methods described in Sections IV to VI were implemented on a VAX 11/780 in the "C" programming language. The CPU times recorded were provided by the "TIME" command built into the UNITY operating system. The images used are 256 × 256 surfels with a resolution of surfels approximately 1.2 mm.

CPU Time Requirements

The processing is divided into three stages for more informative analysis of CPU time required:

1. simultaneous smoothing and mapping the image to gradient space
2. planar surface extraction
3. model building and matching

Stage 1 is computationally very costly. For each image the CPU time required is 1 min 59 s.

Time required for stage 2 varies with the content of the image. The CPU time was recorded for the image in Fig. 5a. The time required is 2 min 14 s, which was one of the most costly images used in the testing.

For stage 3, timing also depends largely on image content. The time for model construction and matching for the image in Fig. 5a is 24.8 s, or about 4.1 s for each object. All of these times include I/O for the image.

The total average processing time per image is approximately 3.5 min. This is a considerable distance from real-time object recognition. Speed was not a major consideration in the development of Witness. The concept of "flexible hardware" (17) for real-time low level image analysis should all but eliminate the first two, most costly stages. Another alternative to this low level processing is to build the object models directly from

the environment, using a programmable laser scanner, as opposed to collecting an entire scene. This option will require intelligent interrogation of the environment, and is presently being investigated at NRC. The model construction and matching stage could easily lend itself to a multiprocessor implementation, with each model to be matched executing on a separate processor.

Image Features Detected

In the simple polyhedral domain, the image features such as surfaces are relatively easy to measure. Since all features in our test images were detected, the only figures that can be given as a measure of performance are the smallest features found in the domain.

The danger of region merging methods such as the one described in Section V is that they tend to merge or grow further than they are intended. The measure of a good region detection method in range images is the angle between planar surfaces which the method can detect. The smallest dihedral angle such as this in the test images used is approximately 20°. The smallest surface is approximately 180 mm^2 in area.

Object Recognition Statistics

Currently, the assumption is made that all objects seen will have a corresponding view model somewhere in the knowledge base. This could easily be changed to a required minimum threshold for a match between two models. Our results would not have improved by this addition for this particular domain.

For the purpose of testing the Witness system, 12 models were constructed, each from a different image, and placed in the knowledge base. Nine images were "shown" to the system containing a total of 38 randomly placed (but not touching) objects. Thirty-four of the 38 were successfully recognized. This is a success rate of about 89.5%. Of the four that failed, three of them were caused by the same error made by Witness. A particular object was matched to the wrong model on all three sightings in the test images. This is because the models for these two objects are virtually identical. Some additions to the view model have been proposed to correct this, but none have been implemented to date. The remaining error is perhaps more serious. A small erroneous surface was detected between two major surfaces. Although the matching algorithm is designed to minimize the effect of the smaller surfaces, this particular error changed the surface adjacency graph in such a way that the two major surfaces were not considered adjacent. It is possible that fine tuning of the thresholds used for surface detection may prevent this from happening. Another possible solution is to add more complexity to the matching algorithm so that it could skip unreliable nodes if a considerably higher score would be the result.

VIII. EXTENSIONS AND CONCLUSIONS

There are many possible extensions for the implementation described in this paper. The constraint that objects may not touch or occlude one another should be removed. A method for this is discussed by Oshima and Shirai (16). They allow for partial occlusion of objects and analyze what is seen as well as what is expected in the occluded area. This method will eventually fail depending on the amount of occlusion, and some alternate method will have to be used. The problem is exacerbated if it is impossible to determine when the method has failed.

Since the world is not constructed of polyhedra, it will eventually be necessary to remove this constraint also. Several methods of doing this have been proposed. One solution is to approximate all surfaces with planar patches. This idea cannot be considered in conjunction with the matching methods described herein. If a curved surface is approximated as a large number of planar surfaces, the graph matching algorithm will become unsuitable for matching models of such complexity. A more appropriate solution is to detect and model the curved surfaces, as is done by Agin and Binford (27). The industrial environment will almost always provide regularly curved surfaces. With range images of the quality provided by our new range finder, it should be feasible to model these curved surfaces with high accuracy and reliability.

Our objective has been to design an approach to computational vision specifically for range images that eludes some of the factors that have prevented widespread use of intelligent robots. Witness provides a new modeling method, low level algorithms to create the models, and high level model matching techniques. We can conclude that object recognition from a high quality single view range image can be successful in a constrained domain/environment. Although the models constructed by Witness are two-dimensional, enriched with three-dimensional information, it is quite possible to correct the projected area, perimeter, etc., to provide true shape of the visible surfaces using these range images. This approach could be used when matching image features to a CAD/CAM type object model. The success rate for object recognition in the constrained domain/environment is quite encouraging, and research is continuing at NRC to improve performance at each stage.

REFERENCES

1. R.A. Jarvis. A perspective on range finding techniques for computer vision. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5: 122-139; 1983.

2. A.K. Mackworth. Interpreting pictures of polyhedral scenes. Proc. 3rd Int. Joint Conf. Artificial Intell. 1973. pp. 556-563.

3. D.G. Morgenthaler and A. Rosenfeld. Surfaces in three dimensional images. Inf. Control, 51: 227-247; 1981.

4. D. Nitzan, A. Brain, and R. Duda. The measurement and use of registered reflectance and range data in scene analysis. Proc. IEEE, 65(2): 206-220; 1977.

5. A Mitiche and J.K. Aggarwal. Detection of edges using range information. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5(2): 174-178; 1982.

6. S. Inokuchi, T. Nita, F. Matsuda, and Y. Sakurai. A three dimensional edge-region operator for range pictures. Proc. 6th Int. Joint Conf. on Pattern Recognition, Munich, Germany. October 1982. pp. 918-920.

7. B. Gil, A. Mitichi, and J.K. Aggarwal. Experiments combining intensity and range edge maps. Comput. Vision, Graphics, Image Process. 21: 395-411; 1983.

8. R. Duda, D. Nitzan, and P. Barett. Use of range and reflectance data to find planar surface regions. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1(3): 259-271; 1979.

9. M. Hebert and J. Ponce. A new method for segmenting 3-D scenes into primitive. Proc. 6th Int. Joint Conf. on Pattern Recognition. October 1982. pp. 836-838.

10. T.C. Henderson and B. Bhanu. Three point seed method for extraction of planar faces from range data. Conference Record of the 1982 Workshop on Industrial Applications of Machine Vision. 1982. pp. 181-186.

11. J. Glicksman. A cooperative scheme for image understanding using multiple sources of information. University of British Columbia Technical Report, TN 82-13. 1982.

12. L.G. Roberts. Machine perception of three dimensional solids. Optical and Electrooptical Information Processing. Edited by A. Vanderburgh. MIT Press, Cambridge, Mass. 1965. pp. 159-167.

13. R. Nevatia and T.O. Binford. Description and recognition of curved objects. Artif. Intell. 8(1): 77-88; 1977.

14. R.J. Popplestone, C.M. Brown, C.M. Ambler, and G.F. Crawford. Forming models of plane-and-cylinder faceted bodies from light stripes. Proc. of 4th IJCAI. 1975. pp. 664-668.

15. G. Falk. Interpretation of line data as a three dimensional scene. Artif. Intell. 3(2): 101-144; 1972.

16. M. Oshima and Y. Shirai. Object recognition using three-dimensional information. IEEE Trans. Pattern Anal. Machine Intell. PAMI-5(4): 353-361; 1983.

17. Y. Shirai. On application of three dimensional computer vision. Bull. Electrotech. Lab. (Japan), 43(6): 358-377; 1979.

18. M. Rioux. Compact, high speed 3-D vision sensor for robotic applications. Proc. 2nd Canadian CAD/CAM and Robotics Conference, Toronto, Ont. 1983. pp. 4.11-4.16.

19. M. Rioux. Laser range finder based on synchronized scanners. Appl. Opt. 23(21): 3837-3844; 1984. NRC 24529.

20. A.K. Mackworth. Use of constraints for interpreting three dimensional scenes. Ph.D. Thesis, University of Sussex, England. 1974.

21. S.A. Shafer, T. Kanade, and J. Kender. Gradient space under orthography and perspective. IEEE 1982 Workshop on Computer Vision: Representation and Control. August 1982. pp. 26-34.

22. D. Marr and E. Hildreth. Theory of edge detection. Massachusetts Institute of Technology AI Memo No. 518. 1979.

23. D.H. Ballard and C.M. Brown. Computer Vision. Prentice Hall, Englewood Cliffs, NJ. 1982.

24. A.A.G. Requicha. Representations for rigid solids: Theory, methods, and systems. Comput. Surv. 12(4): 437-464; 1980.

25. M.R. Garey and D.S. Johnson. Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman and Col., San Francisco, CA. 1979.

26. J.D. Boissonnant and F. Germain. A new approach to the problem of acquiring randomly oriented workpieces out of a bin. Proc. 7th IJCAI. August 1981. pp. 796-802.

27. G.J. Agin and T.O. Binford. Computer description of curved objects. Proc. 3rd IJCAI. August 1973. pp. 629-640.